

# Families with infants: speeding up algorithms for NP-hard problems using FFT\*

Alexander Golovnev <sup>†</sup>Alexander S. Kulikov <sup>‡</sup>Ivan Mihajlin <sup>§</sup>

## Abstract

Assume that a group of people is going to an excursion and our task is to seat them into buses with several constraints each saying that a pair of people does not want to see each other in the same bus. This is a well-known coloring problem and it can be solved in  $O^*(2^n)$  time by the inclusion-exclusion principle as shown by Björklund, Husfeldt, and Koivisto in 2009. Another approach to solve this problem in  $O^*(2^n)$  time is to use the fast Fourier transform. A graph is  $k$ -colorable if and only if the  $k$ -th power of a polynomial containing a monomial  $\prod_{i=1}^n x_i^{[i \in I]}$  for each independent set  $I \subseteq [n]$  of the graph, contains the monomial  $x_1 x_2 \dots x_n$ .

Assume now that we have additional constraints: the group of people contains several infants and these infants should be accompanied by their relatives in a bus. We show that if the number of infants is linear then the problem can be solved in  $O^*((2 - \epsilon)^n)$  time. We use this approach to improve known bounds for several NP-hard problems (the traveling salesman problem, the graph coloring problem, the problem of counting perfect matchings) on graphs of bounded average degree, as well as to simplify the proofs of several known results.

## 1 Introduction

In this paper we consider algorithms for three classical hard problems: the traveling salesman problem, the chromatic number problem, and the problem of counting perfect matchings.  $O^*(2^n)$  algorithms by [4] and [16] for the traveling salesman problem have been known for more than 50 years already ( $n$  is the number of vertices of an input graph,  $O^*$  hides polynomial factors of input length). The upper bound  $O^*(2^n)$  for the chromatic number problem is proved by [10]. The number of perfect matchings can be computed in time  $O^*(2^{n/2})$  as shown by [6] (this matches the bound by [22] for bipartite graphs).

For all three problems mentioned above (chromatic number, traveling salesman, counting perfect matchings), improving the known bounds for the general case is a major open problem in the field of algorithms for NP-hard problems. Better upper bounds are known however for various special cases. For Hamiltonian cycle problem, [5] and [8] proved an  $O(1.66^n)$  bound for the symmetric case (i.e., undirected graphs), [11] proved an  $O^*(1.89^n)$  bound for directed bipartite graphs. In [7, 9, 13]

---

\*This paper is based on the same results as [15], but the presentation of the results and the whole discuss have been reworked substantially. Research is partially supported by the Government of the Russian Federation (grant 14.Z50.31.0030).

<sup>†</sup>New York University

<sup>‡</sup>St. Petersburg Department of Steklov Institute of Mathematics of the Russian Academy of Sciences

<sup>§</sup>St. Petersburg Department of Steklov Institute of Mathematics of the Russian Academy of Sciences

better upper bounds are proven for graphs of bounded degree (the three considered problems are known to be NP-hard even on graphs of bounded degree).

We present a new approach to get bounds of the form  $O^*((2 - \epsilon)^n)$  in various special cases. Namely we show that such a bound follows almost immediately if the corresponding partition problem possesses a certain structure. Informally, this structure can be described as follows. Assume that a group of people is going to an excursion and our task is to seat them into buses with several constraints each saying that a pair of people does not want to see each other in the same bus. This is the coloring problem and it can be solved in  $O^*(2^n)$  time using the inclusion-exclusion method as shown by [10]. Another approach to solve this problem in  $O^*(2^n)$  time is to use the fast Fourier transform (FFT). A graph is  $k$ -colorable if and only if the  $k$ -th power of a polynomial containing a monomial  $\prod_{i=1}^n x_i^{[i \in I]}$  for each independent set  $I \subseteq [n]$  of the graph contains the monomial  $x_1 x_2 \dots x_n$ . This method is mentioned by [12]; previously, [18] and [8] used multilinear monomial detection for solving parameterized problems. Assume now that we have additional constraints: the group of people contains several infants and these infants should be accompanied by their relatives in a bus. Roughly, we prove that if the number of infants is linear then the problem can be solved in  $O^*((2 - \epsilon)^n)$  time using FFT.

Using this approach we unify several known results of this kind. An additional advantage of the approach is the simplicity of using it as a black box. Namely, all one needs to do is to reveal the corresponding structure of families with infants. This way, some of the known upper bounds for the above mentioned problems on graphs of bounded degree follow just in a few lines. By using additional combinatorial ideas we also prove the following new results.

For the chromatic number problem, [9] presented an algorithm running in time  $O^*((2 - \epsilon(\Delta))^n)$  on graphs of bounded maximum degree  $\Delta = O(1)$ . The algorithm is based on Yate's algorithm and Möbius inversion and thus uses exponential space. We extend this result to a wider class of bounded average degree graphs. This closes an open problem concerning the existence of such an algorithm stated by [13].

For the traveling salesman problem on graphs of maximum degree  $\Delta = O(1)$ , [7] presented an algorithm running in time  $O^*((2 - \epsilon(\Delta))^n)$  and exponential space. [13] extended the result to graphs of bounded average (rather than maximum) degree. Both algorithms are based on dynamic programming and the savings in the running time comes from an observation that in case of bounded degree graphs an algorithm does not need to go through all possible subsets of vertices (e.g., a disconnected subgraph does not have a Hamiltonian path for sure). It is also because of the dynamic programming technique that both mentioned algorithms use exponential space. We further extend these results presenting an algorithm running in time  $O^*(M(2 - \epsilon(d))^n)$  and polynomial space on directed graphs of average degree  $d$  with integral weights bounded by  $M$ .

[13] developed an algorithm with running time  $O^*((2 - \epsilon(d))^{n/2})$  and exponential space for counting perfect matching in graphs of average degree  $d$ . We present an algorithm solving this problem in  $O^*((2 - \epsilon(d))^{n/2})$  time and polynomial space. Several bounds of this kind are already known for bipartite graphs [2, 3, 23, 21, 17, 13].

## 2 Notation

Let  $G = (V, E)$  be a simple undirected graph. Throughout the paper we implicitly assume that the set of vertices of a graph under consideration is  $V = \{1, 2, \dots, n\}$ . For simplicity, we consider undirected graphs only (whether a graph is directed or not is only important for the traveling

salesman problem; the presented algorithm works for both undirected and directed graphs).

By  $d(G)$  and  $\Delta(G)$  we denote the average and the maximum degree of  $G$  (we omit  $G$  if it is clear from the context).  $N_G(v)$  is a *neighborhood* of  $v$  in  $G$ , i.e., all the neighbors of  $v$  in  $G$ , and  $N_G[v] = N_G(v) \cup \{v\}$  is its *closed neighborhood*. For  $S \subseteq V$ , by  $G[S]$  we denote a subgraph of  $G$  induced by  $S$ . We use  $G \setminus S$  as a shortcut for  $G[V \setminus S]$ .

The *square* of  $G = (V, E)$  is a graph  $G^2 = (V, E')$  where  $E' \supseteq E$  is

$$E' = \{(u, v) : \text{there is a path of length at most 2 from } u \text{ to } v \text{ in } G\}.$$

Note that  $\Delta(G^2) \leq (\Delta(G))^2$  and hence one can easily find an independent set of size  $\frac{n}{(\Delta(G))^2+1}$  in  $G^2$ .

Following [13], by  $V_{>c}$  we denote a subset of vertices  $V$  of degree greater than  $c$ .  $V_{<c}$ ,  $V_{=c}$ ,  $V_{\leq c}$ ,  $V_{\geq c}$  are defined similarly. By  $\mathbb{Z}_{\geq c}$  we denote the set of all integers greater than or equal to  $c$ .

For an positive integer  $k$ , by  $[k]$  we mean the set of all positive integers less than or equal to  $k$ . While for a Boolean expression  $P$ , by  $[P]$  we mean 1 if  $P$  is true and 0 otherwise (this is the standard Iverson bracket notation).

Throughout the paper by  $\varepsilon$  we denote a positive constant that does not depend on the size of a graph.

### 3 Toolkit

This section describes the main toolkit for proving upper bounds for NP-hard problems using FFT. In the first two subsections we provide the main such tools without proofs and give short proofs of several recently proved upper bounds using these tools. All the proofs are given in the third subsection.

#### 3.1 FFT

In this subsection we remind and adjust the FFT technique for our goals. In the first two theorems we deal with univariate polynomials, and then we proceed to multivariate polynomials.

**Theorem 1.** *Let  $P(x) = \sum_{i=0}^n p_i x^i$ ,  $Q(x) = \sum_{i=0}^n q_i x^i$  be polynomials of degree at most  $n$  with non-negative integer coefficients less than  $W$ . If  $P(x)$  and  $Q(x)$  are given as lists  $(p_0, \dots, p_n)$  and  $(q_0, \dots, q_n)$  of coefficients then the list of coefficients of their product  $S(x) = P(x) \cdot Q(x)$  can be found in time and space  $n \text{ polylog}(n, W)$ .*

**Theorem 2.** *Let  $P(x) = \sum_{i=0}^n p_i x^i$  be a polynomial of degree at most  $n$  with non-negative integer coefficients less than  $W$ . Given an arithmetic circuit  $C(x, p)$  of size  $\text{polylog}(n, W)$  which evaluates  $P$  modulo a prime  $p = O(n \text{ polylog}(n, W))$  at an integer point  $x$ , any coefficient of  $P(x)$  can be found in time  $n \text{ polylog}(n, W)$  and space  $\text{polylog}(n, W)$ .*

We use the previous two theorems to prove the following result dealing with multivariate polynomials.

**Theorem 3.** *Let  $P_1(x_1, \dots, x_n, z), \dots, P_n(x_1, \dots, x_n, z)$  be polynomials of  $n+1$  variables with non-negative integer coefficients less than  $W$ , where individual degrees of  $x$ 's are at most  $n$ , and the degree of  $z$  is at most  $d$ . Let also  $\Pi(x_1, \dots, x_n, z) = \prod_{i=1}^n P_i$  be the product of the polynomials. The smallest  $k \leq nd$ , s.t.  $\Pi$  contains the monomial  $x_1 \dots x_n z^k$ , and the coefficient of this monomial in  $\Pi$  can be found*

1. in time  $d2^n \text{poly}(n, \log W)$  and space  $2^n \text{poly}(n, \log(W, d))$ , if  $P_i$ 's are given as lists of monomials and the length of each list is at most  $2^n$ ;
2. in time  $d2^n \text{poly}(n, \log W)$  and space  $\text{poly}(n, \log(W, d))$ , if  $P_i$ 's are given as circuits  $C_i(x, p)$  of size  $\text{poly}(n, \log(W, d))$  where each  $C_i$  evaluates  $P_i$  modulo a prime  $p = O(d2^n \text{poly}(n, \log W))$  at an integer point  $x$ .

**Remark 1.** In the following, we will use Theorem 3 with  $k \leq n$  polynomials. For this, it is enough to set  $P_{k+1} \equiv P_{k+2} \equiv \dots \equiv P_n \equiv 1$ .

To show the usefulness of Theorem 3 we reprove the following recent result by [20].

**Lemma 1.** The traveling salesman problem on graphs with integer weights from  $[M]$  can be solved in time  $O^*(M2^n)$  and space  $\text{poly}(n, \log M)$ .

Before proving this lemma, we state a technical fact that will be used again further in the text.

**Lemma 2.** For a graph  $G$ , let  $P(x_1, \dots, x_n, z)$  be a polynomial defined as follows:

$$P(x_1, \dots, x_n) = \sum_{\text{closed walk } 1 \rightarrow i_2 \rightarrow \dots \rightarrow i_n \rightarrow 1 \text{ of weight } w} x_1 x_{i_2} \dots x_{i_n} z^w. \quad (1)$$

If the edge weights of  $G$  are from  $[M]$ , then  $P \bmod p$  where  $p = O^*(M2^n)$  can be evaluated in time and space  $\text{poly}(n, \log M)$  at any input  $x_1, \dots, x_n, z < p$ .

*Proof.* The polynomial can be evaluated by the standard dynamic programming. Namely, let  $Q_{i,k}(x_1, \dots, x_n, z)$  be a polynomial containing all walks of length  $k$  starting at the vertex 1 and ending at the vertex  $i$ . The polynomials  $Q_{i,k}$  can be evaluated recursively using a straightforward relation for  $k > 1$ :

$$Q_{i,k}(x_1, \dots, x_n, z) = \sum_{(j,i) \in E(G)} Q_{j,k-1}(x_1, \dots, x_n) \cdot x_i \cdot z^{w(j,i)}.$$

The initial setting is

$$Q_{i,0}(x_1, \dots, x_n, z) = x_i.$$

Then,

$$P(x_1, \dots, x_n, z) = \sum_{(i,1) \in E(G)} Q_{i,n-1}(x_1, \dots, x_n) \cdot z^{w(i,1)}.$$

It is easy to see that the degree of  $z$  in  $P$  is at most  $nM$ , and that the evaluation of  $P \bmod p$  requires only  $\text{poly}(n, \log M)$  time and space.  $\square$

*Proof of Lemma 1.* All one needs to do is to find the smallest  $k$  such that the polynomial  $P$  defined by (1) contains the monomial  $x_1 \dots x_n z^k$ . By Lemma 2,  $P$  can be evaluated in  $\text{poly}(n, \log M)$  time and space. Now, the statement follows from the second part of Theorem 3 with  $P_1 = P$  and  $P_2 \equiv \dots \equiv P_n \equiv 1$ .  $\square$

In the next subsection we show also how to use Theorem 3 to find the chromatic number of a graph in time and space  $O^*(2^n)$ .

### 3.2 Partition problems and families with infants

**Definition 1** (partition problem). *Let  $1 \leq k \leq n$  be integers and let  $\mathcal{F} = \{\mathcal{F}_1, \dots, \mathcal{F}_k\}$  where each  $\mathcal{F}_i \subseteq 2^{[n]}$  is a family of subsets of  $[n]$ . An  $(n, k, \mathcal{F})$ -partition problem is to represent  $[n]$  as a disjoint union of  $k$  sets from  $\mathcal{F}_i$ 's:  $[n] = F_1 \sqcup \dots \sqcup F_k$ , where  $F_i \in \mathcal{F}_i, \forall 1 \leq i \leq k$ .*

This definition is similar to the one used by [10], the only difference being that in the definition above the families  $\mathcal{F}_i$ 's are not necessarily equal. The brute force search algorithm for this problem takes time  $O^*(\max_{1 \leq i \leq k} |\mathcal{F}_i|^k)$ . Using FFT one can easily prove an upper bound  $O^*(2^n)$  which beats the previously mentioned bound in many interesting cases.

There exists a natural one-to-one mapping between families of subsets of  $[n]$  and zero-one multilinear polynomials of  $n$  variables: for  $\mathcal{F} \subseteq 2^{[n]}$ , denote by  $P_{\mathcal{F}}$  the following polynomial:

$$P_{\mathcal{F}}(x_1, \dots, x_n) = \sum_{F \in \mathcal{F}} \prod_{i=1}^n x_i^{[i \in F]}.$$

I.e., elements of  $\mathcal{F}$  correspond to monomials of  $P_{\mathcal{F}}$ . Conversely, for a zero-one multilinear polynomial  $P$  we denote by  $\mathcal{F}_P$  the corresponding family of subsets of  $[n]$ . There is also a natural extension of the function  $\mathcal{F}_P$  for all polynomials (but not just 0-1 multilinear polynomials):  $\mathcal{F}_P$  contains all subsets  $S \subseteq [n]$  for which  $P$  contains a monomial whose variable set is exactly  $S$  (thus, the coefficient of this monomial must be non-zero and for each  $i \in S$  the degree of  $x_i$  in this monomial must be at least 1). For a monomial  $m$ , by  $F(m)$  we denote the corresponding subset of  $[n]$  and by  $\deg(m)$  we denote the total degree of  $m$ . (E.g., for  $m = x_2^4 x_3 x_9^2$ ,  $F(m) = \{2, 3, 9\}$  and  $\deg(m) = 7$ ).

There is a straightforward reduction from the partition problem to the multilinear monomial detection problem.

**Corollary 1.** *There is a solution for an  $(n, k, \mathcal{F})$ -partition problem if and only if the polynomial  $\prod_{i=1}^k P_{\mathcal{F}_i}$  contains the monomial  $x_1 \dots x_n$ .*

Combined with the first part of Theorem 3 this gives us the following useful corollary.

**Corollary 2.** *Let  $\mathcal{F} = \{\mathcal{F}_1, \dots, \mathcal{F}_k\}$  be a family of subsets of  $[n]$  given as a list. Then the  $(n, k, \mathcal{F})$ -partition problem can be solved in  $O^*(2^n)$  time and space.*

Note that Corollary 2 immediately implies  $O^*(2^n)$  upper bound for such problems as domatic number and chromatic number. These bounds were proved relatively recently by [10] using the inclusion-exclusion method.

Below, we formally define a combinatorial structure called families with infants that allows to prove stronger than  $O^*(2^n)$  upper bounds.

**Definition 2** (families with infants for subsets). *Let  $\mathcal{S} \subseteq 2^{[n]}$ .*

$\mathcal{R} = ((R_1, r_1), \dots, (R_p, r_p))$  *is called a  $(p, q)$ -system of families with infants for  $\mathcal{S}$  if all of the following conditions are satisfied:*

1. *for all  $i = 1, \dots, p$ :  $r_i \in R_i \subseteq [n]$ ;  $r_i$  is called an infant and all the elements of  $R_i \setminus \{r_i\}$  are called relatives of  $r_i$ ; the sets  $R_i$  are called families;*
2. *the size of each family  $R_i$  is at most  $q$ ;*
3.  *$pq \leq n$ ;*

4. all families  $R_i$ 's are pairwise disjoint:  $R_i \cap R_j = \emptyset$  for  $i \neq j$ ;
5. in any set  $S$  of  $\mathcal{S}$  each infant is accompanied by at least one of its relatives:

$$\forall S \in \mathcal{S}, \text{ if } r_i \in S \text{ then } |F \cap R_i| \geq 2. \quad (2)$$

**Definition 3** (families with infants for partitions problems).  $\mathcal{R}$  is called a system of families with infants for an  $(n, k, \mathcal{F})$ -partition problem where  $\mathcal{F} = (\mathcal{F}_1, \dots, \mathcal{F}_k)$  if it is so for all  $\mathcal{F}_i$ ,  $i \in [k]$ .

**Definition 4** (families with infants for polynomials).  $\mathcal{R}$  is called a system of families with infants for a polynomial  $P$  over  $n$  variables if it is so for  $\mathcal{F}_P$ .

The next theorem constitutes the main technical result of the paper saying that if a problem possesses the property of families with infants then one can solve the corresponding problem in time  $O^*((2 - \varepsilon)^n)$ . This property is particularly easy to show for problems on graphs of bounded degree (either maximum or average).

**Theorem 4.** Let  $P_1, \dots, P_n$  be polynomials of  $x_1, \dots, x_n, z$  as in Theorem 3 (the coefficients are from  $[W]$ ,  $\deg x_i \leq n$ ,  $\deg z \leq d$ ). Let also  $P'_i(x_1, \dots, x_n) = P_i(x_1, \dots, x_n, 1)$ , and let  $\mathcal{R} = ((R_1, r_1), \dots, (R_p, r_p))$  be a  $(p, q)$ -system of families with infants for  $P'_1, \dots, P'_n$ . Then the smallest  $k$  such that  $\Pi = \prod_{i=1}^n P_i$  contains the monomial  $x_1 \dots x_n z^k$  can be found in time

$$O^* \left( d \cdot 2^n \cdot \left( \frac{2^q - 1}{2^q} \right)^p \cdot 2^q \right) \quad (3)$$

1. and the same space, if the polynomials  $P_i$ 's are multilinear in  $x_1, \dots, x_n$  and are given as lists of monomials;
2. and space  $\text{poly}(n, \log W)$  if  $P_i$ 's are given by circuits of size  $\text{poly}(n, \log W)$ .

In particular, if  $q = O(1)$  and  $p = \Omega(n)$  then the upper bound on the running time is  $O^*(d(2 - \varepsilon)^n)$ , where  $\varepsilon$  does not depend on  $n$ .

Roughly, the savings in the running time comes from the fact that while looking for a valid partition of  $[n]$  one can avoid the case  $F \cap R_i = \{r_i\}$  (i.e., instead of considering all  $2^q$  possibilities of  $F \cap R_i$ , one considers  $2^q - 1$  of them).

Theorem 4 also implies the  $O^*((2 - \varepsilon)^n)$  upper bound for a partition problem with a  $(\Omega(n), O(1))$ -system of families with infants.

**Corollary 3.** Given  $\mathcal{F} = \{\mathcal{F}_1, \dots, \mathcal{F}_k\}$  where  $k \leq n$  and  $\mathcal{F}_i \subseteq 2^{[n]}$  as a list of subsets and an  $(\Omega(n), O(1))$ -system of families with infants  $\mathcal{R}$  for the  $(n, k, \mathcal{F})$ -partition problem one can solve this problem in time and space  $O^*((2 - \varepsilon)^n)$ .

As an illustration of using Corollary 3 we replicate a result from [9]. In the (decision version of) domatic number problem the question is to check whether it is possible to partition the set of vertices into  $k$  dominating sets.

**Lemma 3.** The domatic number problem in a graph of maximum degree  $\Delta = O(1)$  can be solved in time and space  $O^*((2 - \varepsilon(\Delta))^n)$ .

*Proof.* The domatic number problem is an  $(n, k, \mathcal{F})$ -problem where each  $\mathcal{F}_i$  is just the set of all dominating sets of  $G$ . By definition, for any  $v \in [n]$  and any dominating set  $U \subseteq [n]$ ,  $N_G[v] \cap U \neq \emptyset$ . This gives a straightforward construction of families with infants.

Find greedily an independent set  $I \subseteq V$  of size  $p = \frac{n}{\Delta^2+2}$  in  $G^2$ . Assume w.l.o.g. that  $I = \{1, \dots, p\}$ . For each  $1 \leq i \leq p$ , let  $R_i = N_G[i]$ . At this point we have at least  $n - p(\Delta + 1) \geq p$  remaining vertices in  $V \setminus \bigcup_{i=1}^p R_i$ . So, we can extend each  $R_i$  by one vertex and declare this one additional vertex as the infant of  $R_i$ .

All  $R_i$  have size at most  $q = \Delta + 2 = O(1)$ , the total number of  $R_i$ 's is  $p = \frac{n}{\Delta^2+2} = \Omega(n)$ . Clearly  $pq \leq n$ . The constructed sets satisfy the property (2) by the following reason. Each  $R_i$  is a proper superset of  $N_G[v]$  for some  $v \in V$ , moreover no vertex from  $N_G[v]$  is the infant of the family  $R_i$ . For any dominating set  $U$  and any vertex  $v$ ,  $U \cap N_G[v] \neq \emptyset$ . Thus, any dominating set contains at least one relative of  $r_i$  (even if it does not contain  $r_i$  itself).

The upper bound now follows from Corollary 3.  $\square$

Another example is a  $O^*((2 - \varepsilon)^n)$  algorithm for the traveling salesman problem for graphs of bounded degree. This result was given by [7].

**Lemma 4.** *The traveling salesman problem on a graph of maximum degree  $\Delta = O(1)$  and with integer weights from  $[M]$  can be solved in time and space  $O^*(M(2 - \varepsilon(\Delta))^n)$ .*

*Proof.* It is enough to find a system of families with infants for the polynomial  $P$  from (1). For this, we just construct greedily an independent set  $I$  of size  $p = \frac{n}{\Delta^2+1}$  in  $G^2$ . Assume that  $I = \{1, \dots, p\}$  and let  $R_i = N_G[i]$ ,  $r_i = i$ . This is clearly a  $(p, q)$ -system of sets with infants for  $q = \Delta + 1$ , since in each closed walk in  $G$  an infant  $i$  must be accompanied by one of its neighbours. Theorem 4 then implies an upper bound  $O^*((2 - \varepsilon(\Delta))^n)$ .  $\square$

### 3.3 Proofs

*Proof of Theorem 1.* Let  $m = 2W^2n+1$ ,  $\mathbb{Z}_m$  be the ring of integers modulo  $m$ . Since each coefficient of  $S(x) = P(x) \cdot Q(x)$  is a positive integer less than  $m$ , it suffices to find  $\bar{S}(x) = \bar{P}(x) \cdot \bar{Q}(x)$ , where  $\bar{P}(x), \bar{Q}(x), \bar{S}(x) \in \mathbb{Z}_m[x]$ . Note that one can perform arithmetic operations in  $\mathbb{Z}_m$  in time  $\text{polylog}(m)$ . Since  $2 \in \mathbb{Z}_m^*$ ,  $P(x)$  and  $Q(x)$  can be multiplied in time  $n \text{ polylog}(n) \text{ polylog}(m) = n \text{ polylog}(n, W)$  (see, e.g., Exercise 17.24 in [24]).  $\square$

*Proof of Theorem 2.* Assume one is to find the coefficient  $p_m$  of the monomial  $x^m$  in  $P(x) = \sum_{i=0}^n p_i x^i$ . By the prime number theorem, there exist at least  $k = \log W$  primes in the interval  $[n, n \text{ polylog}(n, W)]$ . Using a deterministic polynomial primality-testing algorithm (e.g., the AKS algorithm [1]) one can find primes  $q_1, \dots, q_k$  in this interval in time  $n \text{ polylog}(n, W)$ . By the Chinese remainder theorem, it suffices to find  $p_m$  modulo  $q_1, \dots, q_k$  in the desired time and space.

We show how to find  $p_m$  modulo  $q = q_i$ . Since  $q = n \text{ polylog}(n, W)$ , one can factor  $q - 1$  in deterministic time  $n \text{ polylog}(n, W)$  using trivial division. Given the factorization of  $q - 1 = a_1^{b_1} \dots a_s^{b_s}$ , one finds a primitive root of unity in  $\omega \in \mathbb{Z}_q^*$  in  $n \text{ polylog}(n, W)$  steps as follows. For every  $t \in \mathbb{Z}_q^*$ , one checks if  $t^{\frac{q-1}{a_i}} \neq 1$  for all  $i \in [s]$ , this takes  $qs \text{ polylog}(q) = n \text{ polylog}(n, W)$  steps. Now the coefficient  $p_m$  of  $x^m$  in  $\mathbb{Z}_q^*$  equals

$$\frac{1}{q-1} \sum_{i=0}^{q-2} \omega^{-im} P(\omega^i).$$

Since  $P(x)$  can be evaluated in  $\mathbb{Z}_q^*$  in  $\text{polylog}(n, W)$  time and space, one needs  $n \text{ polylog}(n, W)$  time and only  $\text{polylog}(n, W)$  space to find one coefficient of  $P$ .  $\square$

For a subset  $U \subseteq [n]$ , let  $b(U) \in \{0, 1\}^n$  denote the characteristic vector of the set  $U$  (i.e.,  $b(U)[i] = 1$  if and only if  $i \in U$ ). In the analysis below we sometimes identify a bit vector  $b(U)$  with a non-negative integer between 0 and  $2^n - 1$  that it represents.

For a bit vector  $b$ , we denote the Hamming weight of  $b$ , i.e., the number of 1's in  $b$ , by  $w(b)$ . Note the following simple fact: for any two non-negative integers  $a$  and  $b$ ,

$$w(\text{bin}(a)) + w(\text{bin}(b)) \geq w(\text{bin}(a + b)) \quad (4)$$

and the equality holds if and only if there are no carries in  $a + b$ .

**Lemma 5.** *Let  $T_1(x_1, \dots, x_n), \dots, T_k(x_1, \dots, x_n)$  be polynomials of  $n$  variables  $x_1, \dots, x_n$  with non-negative coefficients. Let also*

$$Q_i(x, y) = \sum_{m \in T_i} y^{\deg(m)} x^{b(F(m))}.$$

*Then  $T = \prod_{i=1}^k T_i(x_1, \dots, x_n)$  contains the monomial  $x_1 \dots x_n$  if and only if  $Q(x, y) = \prod_{i=1}^k Q_i(x, y)$  contains the monomial  $y^n x^{b([n])}$ .*

*Proof.* One direction of this statement is straightforward. If  $T$  contains the (multilinear) monomial  $x_1 \dots x_n$ , then  $T_1, \dots, T_k$  contain multilinear monomials  $m_1, \dots, m_k$  such that  $m_1 \dots m_k = x_1 \dots x_n$ . Then  $F(m_1) \sqcup \dots \sqcup F(m_k) = [n]$  and

$$\prod_{i=1}^k y^{\deg(m_i)} x^{b(F(m_i))} = y^n x^{b([n])}.$$

For the reverse direction, assume that  $Q = \prod Q_i$  contains the monomial  $y^n x^{b([n])}$ . Because of the term  $y^n$ , there exist  $k$  monomials  $m_1 \in T_1, \dots, m_k \in T_k$  such that  $\deg(m_1) + \dots + \deg(m_k) = n$ . In other words, the total number of 1's in all characteristic vectors of  $F(m_i)$ 's at most  $n$ . Moreover,

$$b(F(m_1)) + \dots + b(F(m_k)) = b([n]). \quad (5)$$

From (4), one concludes that the equality (5) is only possible when

$$w(b(F(m_1))) + \dots + w(b(F(m_k))) = n$$

and there are no carries in (5). This in turn implies that  $\{F(m_1), \dots, F(m_k)\}$  is a partition of  $[n]$ .  $\square$

*Proof of Theorem 3.* Let for  $1 \leq i \leq n$ ,  $P_i(x_1, \dots, x_n, z) = \sum_{j=0}^d z^j T_{ij}(x_1, \dots, x_n)$ , where  $T_{ij}$ 's are multilinear polynomials. Let also

$$Q_i(x, y, z) = \sum_{j=0}^d z^j \sum_{m \in T_{ij}} y^{\deg(m)} x^{b(F(m))}.$$

From Lemma 5,  $\Pi = \prod P_i$  contains the monomial  $x_1 \dots x_n z^k$  if and only if  $Q = \prod Q_i$  contains the monomial  $z^k y^n x^{b([n])}$ .



Now it suffices to show how to efficiently find the sum of coefficients of the monomials  $z^i y^n x^{b([n])}$  for  $i \leq k$ . Indeed, since all the coefficients are positive, the binary search on  $k$  gives us the smallest  $k$  s.t.  $\Pi$  contains  $x_1 \dots x_n z^k$ , and the coefficient of this monomial in  $\Pi$ .

It is easy to see that the degree of  $z$  in  $Q = \prod_{i=1}^n Q_i(x, y, z)$  does not exceed  $dn$ . Similarly, the degree of  $y$  does not exceed  $n^2$ . Therefore, in order to obtain univariate polynomials we can use Kronecker substitution [19]. Namely, we replace  $y$  by  $z^{dn+1}$ , and  $x$  by  $z^{(dn+1)(n^2+1)}$ . Thus, for each  $1 \leq i \leq n$  we consider a univariate polynomial  $Q'_i(z)$ :

$$Q'_i(z) = \sum_{j=0}^d z^j \sum_{m \in T_{ij}} z^{(dn+1)|m|} z^{(dn+1)(n^2+1)b(F(m))}.$$

It is easy to see that the coefficient of  $z^{a_1+(dn+1)a_2+(dn+1)(n^2+1)a_3}$  (where  $a_1 \leq dn, a_2 \leq n^2$ ) in  $Q'(z) = \prod Q'_i(z)$  equals the coefficient of  $z^{a_1} y^{a_2} x^{a_3}$  in  $Q = \prod Q_i(x)$ . In other words, we associate an integer from  $[0..(dn+1)(n^2+1)2^n]$  with each monomial  $z^i m$ , where  $m$  is a monomial over  $\{x_1, \dots, x_n\}$ . This integer is an encoding of  $i$  and  $F(m)$  in  $n+3\log n + \log d$  bits, s.t. the first  $n$  bits indicate elements of  $F(m)$ , the next  $\log n$  bits are zeros, then  $\log n$  bits are the binary expansion of  $|m|$ ,  $\log n$  zeros again, and the last  $\log d$  bits encode  $i$ . We need to find the sum of coefficients of

$$z^{i+n(nd+1)+(nd+1)(n^2+1)b([n])}$$

in  $Q'(z) = \prod_{i=1}^k Q'_i(z)$  for  $i \leq k$ . In order to do that, we multiply  $Q'$  by  $(1+z+\dots+z^k) = (z^{k+1}-1)/(z-1)$ . The coefficient of

$$z^{k+n(nd+1)+(nd+1)(n^2+1)b([n])}$$

in the obtained polynomial  $Q''(z) = (z^{k+1}-1)/(z-1)Q'(z)$  is equal to the sum of coefficients of

$$z^{i+n(nd+1)+(nd+1)(n^2+1)b([n])}$$

in  $Q'(z)$  for all  $i \leq k$ .

Note that the degree of  $Q''(z)$  is  $O(d2^n \text{poly}(n))$ . Now Theorem 1 and Theorem 2 finish the proof. Namely, to prove (1), we just apply the FFT from Theorem 1  $(n-1)$  times. In order to prove (2), we note that  $Q''(z) \bmod p$  can be easily obtained from the values  $P_i(x_1, \dots, x_n, z) \bmod p$ .  $\square$

**Definition 5.** For a matrix  $M = (M[i, j])_{0 \leq i \leq p-1, 0 \leq j \leq q-1} \in \mathbb{Z}_{\geq 0}^{p \times q}$  let

$$\begin{aligned} \text{colweight}(M, j) &= \sum_{i=0}^{p-1} M[i, j], \text{weight}(M) = \sum_{i=0}^{p-1} \sum_{j=0}^{q-1} M[i, j] = \sum_{j=0}^{q-1} \text{colweight}(M, j), \\ \text{rowcode}(M, i) &= -M[i, 0] + \sum_{j=1}^{q-1} 2^j \cdot M[i, j], \text{rowsum}(M) = \sum_{i=0}^{p-1} \text{rowcode}(M, i), \\ \text{code}(M) &= \sum_{i=0}^{p-1} (2^q - 1)^i \cdot \text{rowcode}(M, i). \end{aligned}$$

**Definition 6.** A matrix  $M \in \mathbb{Z}_{\geq 0}^{p \times q}$  is called row-normalized if  $\text{rowcode}(M, i) \geq 0$  for all  $0 \leq i \leq p-1$ .

**Remark 2.** In the analysis below we will need the following simple estimates. Let  $E \in \{0, 1\}^{p \times q}$ . Then

1.  $\text{rowcode}(E, i) < 0$  if and only if  $E[i] = [1, 0, 0, \dots, 0]$ ,
2.  $\text{rowcode}(E, i) \leq 2^q - 2$ .
3.  $\text{code}(E) \leq (2^q - 2) \cdot \sum_{i=0}^{p-1} (2^q - 1)^i < (2^q - 1)^p$  (assuming  $q \geq 2$ ).

The following fact is well known so we state it without a proof.

**Lemma 6.** The expansion of  $X \in \mathbb{Z}_{\geq 0}$  in powers of  $b > 1$  as  $X = \sum_{i=0}^{\infty} x_i \cdot b^i, x_i \geq 0$  has the minimal value of the sum of digits  $\sum_{i=0}^{\infty} x_i$  if and only if  $\forall i : 0 \leq x_i < b$  (i.e.,  $X$  is written in the numeral system of base  $b$ ).

**Lemma 7.** Let  $q \geq 2$  and  $E \in \{0, 1\}^{p \times q}$  and  $M \in \mathbb{Z}_{\geq 0}^{p \times q}$  be row-normalized matrices. If  $\text{colweight}(M, 0) = \text{colweight}(E, 0)$ ,  $\text{weight}(M) = \text{weight}(E)$ ,  $\text{rowsum}(M) = \text{rowsum}(E)$ ,  $\text{code}(M) = \text{code}(E)$ , then  $M = E$ .

*Proof.* The claim follows from Lemma 6. Since for all  $i$ ,  $\text{rowcode}(E, i) \leq 2^q - 2$ ,  $\text{code}(E)$  has the minimal sum of digits in base  $(2^q - 1)$  system. This in turn implies that for each  $i$ ,  $\text{rowcode}(E, i) = \text{rowcode}(M, i)$ . Then the first columns of matrices  $E$  and  $M$  are equal modulo 2, because parities of rowcodes depend only on the first column. Since  $\text{colweight}(M, 0) = \text{colweight}(E, 0)$  we conclude that the first columns of  $M$  and  $E$  are equal.

Now each  $\text{rowcode}(E, i)$  has the minimal sum of digits in the system of base 2, which means that  $\text{weight}(E)$  has the minimal possible value for these rowcodes. It follows from Lemma 6 that each  $M[i, j]$  must be equal to  $E[i, j]$ .  $\square$

**Definition 7.** An injective function  $\alpha : U \rightarrow \{0, \dots, p-1\} \times \{0, \dots, q-1\}$  is called a matrix representation of a set  $U$ . For such  $\alpha$  and  $S \subseteq U$ , a characteristic matrix  $M_\alpha(S) \in \{0, 1\}^{p \times q}$  is defined as follows:  $i \in U$  if and only if  $M_\alpha(S)[\alpha(i)] = 1$ .

*Proof of Theorem 4.* Let  $\mathcal{R} = ((R_1, r_1), \dots, (R_p, r_p))$  be a  $(p, q)$ -system of families with infants for a  $P'_1, \dots, P'_n$ . Append arbitrary elements from  $[n]$  to families so that the size of each family equals  $q$  and the families are still disjoint (this is possible since  $pq \leq n$ ). Denote the union of families by  $R$  and the rest of  $[n]$  by  $L$ . For each family  $R_i$ , fix an order of its elements such that the 0th element is  $r_i$ . Now consider a matrix representation  $\alpha : [n] \rightarrow \{0, \dots, p-1\} \times \{0, \dots, q-1\}$  defined as follows. If  $v$  is the  $j$ th element of  $R_i$ , then  $\alpha(v) = (i, j)$ . We encode each monomial  $m_i \in P_i$  by parts. We encode elements from  $F(m_i) \cap L$  using the standard technique from Lemma 5. To encode elements from  $F(m_i) \cap R$  we use the characteristic matrix  $M_\alpha(F(m_i) \cap R)$ . Note that  $M_\alpha(F(m_i) \cap R)$  is a row-normalized matrix, because if  $F(m_i)$  contains an infant  $r_i$  of a family  $R_i$ , then it must contain at least one other element from the same row. Consider the following polynomials for  $1 \leq i \leq k$ :  $Q_i(u_1, u_2, u_3, u_4, u_5, u_6, z)$  is equal to

$$\sum_{m=z^t m_1 \in P_i, F=F(m_1)} u_1^{|F \cap L|} \cdot u_2^{b(F \cap L)} \cdot u_3^{\text{colweight}(M, 0)} \cdot u_4^{\text{weight}(M)} \cdot u_5^{\text{rowsum}(M)} \cdot u_6^{\text{code}(M)} \cdot z^t,$$

where  $M = M_\alpha(F \cap R)$  and  $b(F \cap L)$  is an integer from 0 to  $2^{|L|} - 1$ .

We claim that  $\prod_{i=1}^n P_i$  contains the monomial  $x_1 \dots x_n z^t$  if and only if  $\prod_{i=1}^n Q_i$  contains the monomial

$$u_1^{|L|} u_2^{b(L)} u_3^{\text{colweight}(R,0)} u_4^{\text{weight}(R)} u_5^{\text{rowsum}(R)} u_6^{\text{code}(R)} z^t.$$

Indeed, as it was shown in Lemma 5,  $u_1^{|L|} u_2^{b(L)}$  corresponds to partitions of  $L$ . Lemma 7 implies that only partitions of  $R$  may have the term

$$u_3^{\text{colweight}(R,0)} u_4^{\text{weight}(R)} u_5^{\text{rowsum}(R)} u_6^{\text{code}(R)}.$$

Note that the degrees of  $u_1, u_3, u_4$  in  $\prod_{i=1}^k Q_i(u_1, u_2, u_3, u_4, u_5, u_6, z)$  are bounded from above by  $n^2$ , the degree of  $u_2$  is bounded by  $n \cdot 2^{|L|}$ , the degree of  $u_5$  is bounded by  $n^2 \cdot 2^q$ , the degree of  $u_6$  is bounded by  $n \cdot (2^q - 1)^p$ , the degree of  $z$  is bounded by  $dn$ . Now we can apply Kronecker substitution as in the proof of Theorem 3:

$$\begin{aligned} u_1 &= u \\ u_2 &= u^{(n+1)^2} \\ u_3 &= u^{(n+1)^3 2^{|L|}} \\ u_4 &= u^{(n+1)^5 2^{|L|}} \\ u_5 &= u^{(n+1)^7 2^{|L|}} \\ u_6 &= u^{(n+1)^9 2^{|L|} 2^q} \\ z &= u^{(n+1)^{10} 2^{|L|} 2^q (2^q - 1)^p} \end{aligned}$$

The running time of FFT is bounded by the degree of the resulting univariate polynomial, i.e.

$$O^*(\text{poly}(n) d 2^{|L|} 2^q (2^q - 1)^p) = O^*(d 2^{n-pq} (2^q - 1)^p 2^q) = O^*\left(d \cdot 2^n \cdot \left(\frac{2^q - 1}{2^q}\right)^p \cdot 2^q\right).$$

The required statements now follow from Theorems 1 and 2. □

## 4 Properties of bounded degree graphs

The following three lemmas are proven by Cygan and Pilipczuk [13]. We slightly extend the statements and provide the proofs for the sake of completeness.

The lemma below allows to find in a graph a set of vertices of high degree with a better upper bound on its size than given by the standard averaging argument.

**Lemma 8** ([13], Lemma 3.2). *For any graph  $G = (V, E)$  of average degree at most  $d$ , any integer  $m \geq 1$  and any  $\alpha \geq 1$  there exists  $m \leq D \leq M$  such that  $|V_{>D}| \leq \frac{nd}{\alpha D}$  where  $M = \lfloor m e^{\alpha+1} + 1 \rfloor$ .*

*Proof.* Clearly,

$$\sum_{i=0}^{\infty} |V_{>i}| = \sum_{i=0}^{\infty} i |V_{=i}| \leq nd.$$

If, on the other hand,  $|V_{>i}| > \frac{nd}{\alpha i}$  for all  $m \leq i \leq M$  then

$$\sum_{i=0}^{\infty} |V_{>i}| \geq \sum_{i=m}^M |V_{>i}| > \frac{nd}{\alpha} \sum_{i=m}^M \frac{1}{i} = \frac{nd}{\alpha} \left( \sum_{i=1}^M \frac{1}{i} - \sum_{i=1}^{m-1} \frac{1}{i} \right) \geq \frac{nd}{\alpha} (\ln M - \ln(em)) \geq nd,$$

where the next to last inequality uses the well-known estimate for the harmonic series:

$$\ln(i+1) \leq 1 + \frac{1}{2} + \dots + \frac{1}{i} \leq \ln i + 1 = \ln(ie).$$

□

**Remark 3.** Such  $D$  can be easily found in polynomial time by going through all the values  $D = m, \dots, M$ .

The next lemma shows that one can find a subset of vertices of linear size that is independent in the square of a graph.

**Lemma 9** ([13], Lemma 3.1). *For any graph  $G = (V, E)$  of average degree at most  $d$  and maximum degree at most  $\Delta$  one can find in polynomial time a subset of vertices  $B \subseteq V$  such that*

1. *the size of  $B$  is linear in the number of vertices:  $|B| \geq \frac{n}{6\Delta d}$ ;*
2. *degrees of vertices from  $B$  are small: for any  $v \in B$ ,  $\deg_G(v) \leq 2d$ ;*
3.  *$B$  is independent in  $G^2$ : for any  $u \neq v \in B$ ,  $N_G[u] \cap N_G[v] = \emptyset$ .*

*Proof.* Clearly,  $|V_{\leq 2d}| \geq \frac{nd}{2d} = \frac{n}{2}$ . The required set  $B$  can be constructed by a straightforward greedy algorithm: while  $V_{\leq 2d}$  is not empty, take any  $v \in V_{\leq 2d}$ , add it to  $B$ , and remove from  $V_{\leq 2d}$  the vertex  $v$  together with all its neighbors in  $G^2$ . The number of such neighbors is at most  $2d + 2d(\Delta - 1) = 2d\Delta$ . Hence at each iteration at most  $2d\Delta + 1$  vertices are removed and the total number of iterations is at least

$$\frac{|V_{\leq 2d}|}{2d\Delta + 1} \geq \frac{n}{4d\Delta + 2} \geq \frac{n}{6d\Delta}.$$

□

The following lemma allows us to find efficiently in a graph  $G$  of average degree  $d = O(1)$  a subset of vertices  $Y$  of high degree such that  $(G \setminus Y)^2$  contains a large enough independent set. The last inequality in the statement can be seen as exponential savings in the running time.

**Lemma 10** ([13], Lemma 3.4). *For any constants  $\nu \geq 1, \mu < 1, a \geq 0, 0 < c < 1$  there exists  $\beta > 0$  such that for any graph  $G = (V, E)$  of average degree  $d = O(1)$  one can find in polynomial time subsets  $A, Y \subseteq V$  such that:*

1.  $A \cap Y = \emptyset$ ;
2.  $A$  is an independent set in  $(G \setminus Y)^2$ ;
3.  $2|Y| \leq |A| \leq cn$ ;

4. each vertex from  $A$  has at most  $2d$  neighbors in  $G \setminus Y$ :  $\forall v \in A, |\{u \in V \setminus Y : (u, v) \in E\}| \leq 2d$ ;

5.

$$\left(\frac{|A|}{|Y|}\right)^a \nu^{|Y|} \mu^{|A|} < 2^{-\beta n}. \quad (6)$$

*Proof.* Let  $\alpha = \alpha(d, \nu, \mu, a, c)$  be a large enough constant to be defined later. Using Lemma 8 we can find

$$\frac{1}{12dc} \leq D \leq \frac{1}{12dc} e^{\alpha+1} + 1, \quad (7)$$

such that  $|V_{>D}| < \frac{nd}{\alpha D}$ . Let  $Y = V_{>D}$ . Note that the graph  $G \setminus Y$  has average degree at most  $d$  and maximum degree at most  $D$ . Lemma 9 allows us to find a subset  $A \subseteq V \setminus Y$  such that  $A$  is independent in  $(G \setminus Y)^2$ , for all  $v \in A$ ,  $\deg_{G \setminus Y}(v) \leq 2d$  and

$$|A| \geq \frac{n - |Y|}{6dD} \geq \frac{n}{12dD},$$

where the last inequality is true when  $\alpha \geq 2d$ , i.e.,  $\alpha$  is large enough. Remove from  $A$  arbitrary vertices such that  $|A| = \frac{n}{12dD}$ . Because of (7),  $\frac{n}{12dD} \leq nc$ . To guarantee that  $|A| > 2|Y|$  it is enough to take  $\alpha \geq 24d^2$ .

We now show how to choose  $\alpha$  such that the last inequality from the statement is satisfied. Using the well known estimates  $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$  and  $\binom{n}{k} \leq \binom{n}{k'}$  for  $k \leq k' \leq \frac{n}{2}$  we get

$$\left(\frac{|A|}{|Y|}\right)^a \leq \left(\frac{en\alpha D}{12dDnd}\right)^{\frac{nda}{\alpha D}} = \left(\frac{e\alpha}{12d^2}\right)^{\frac{nda}{\alpha D}} = (\gamma\alpha)^{\frac{nda}{\alpha D}},$$

where  $\gamma = \frac{e}{12d^2}$  is a constant. Thus we can upper-bound (6) as follows:

$$\left((\gamma\alpha)^{\frac{da}{\alpha}} \left(\nu^d\right)^{\frac{1}{\alpha}} \mu^{\frac{1}{12d}}\right)^{\frac{n}{D}}. \quad (8)$$

Recall now that  $\mu < 1$  and note that  $(\gamma\alpha)^{\frac{da}{\alpha}} \rightarrow 1$  and  $(\nu^d)^{\frac{1}{\alpha}} \rightarrow 1$  with  $\alpha \rightarrow +\infty$ . Thus for a large enough  $\alpha$ , (8) is  $\left(2^{-\beta'}\right)^{\frac{n}{D}}$  for a constant  $\beta' > 0$ . It remains to recall that  $D < e^\alpha$  and take  $\beta = \beta' e^{-\alpha}$ .  $\square$

## 5 The chromatic number problem

**Definition 8.** In the list coloring problem each vertex  $v$  of the input graph is assigned a list  $L_v$  of allowed colors and the task is to properly color a graph such that each vertex is given a color from its list.

To reduce the search space in the list coloring problem we introduce the following problem.

**Definition 9.** In the coloring with preferences problem besides the lists for all the vertices of a graph  $G$  one is given a set  $P$  of pairs  $\{(u_1, v_1), \dots, (u_t, v_t)\}$  of vertices such that

1. all the vertices from  $P$  are different:  $u_i \neq v_j$  for all  $1 \leq i, j \leq t$ ;

2. the set  $\{u_1, v_1, \dots, u_t, v_t\}$  is independent in  $G^2$ ;

3.  $L_{u_i} = L_{v_i}$  for all  $1 \leq i \leq t$ .

The goal is to color the graph properly using an allowed color for each vertex such that for all  $1 \leq i \leq t$  at least one of the vertices from  $N[u_i]$  is assigned the same color as  $v_i$ .

**Lemma 11.** *Let  $G$  be a graph,  $\{L_v\}_{v \in V}$  be a set of list colors for its vertices, and  $P = \{(u_1, v_1), \dots, (u_t, v_t)\}$  be a set as in Definition 9. Then there is a solution for an instance  $(G, \{L_v\})$  of the list coloring problem if and only if there is a solution for an instance  $(G, \{L_v\}, P)$  of the coloring with preferences problem.*

*Proof.* Obviously, if there is a coloring satisfying preferences then it is also a list-coloring. For the reverse direction, consider a proper list coloring such that for some vertex  $u_i$  neither of the vertices from  $N[u_i]$  is given the same color as  $v_i$ . One can then change the color of  $u_i$  to the one of  $v_i$  (this is allowed since  $L_{u_i} = L_{v_i}$ ). This clearly does not violate any coloring constraints and it strictly increases the number of pairs  $(u_i, v_i)$  satisfying its preference constraints. It remains to note that this recoloring cannot violate any other preference constraint since the set of vertices from  $P$  is independent in  $G^2$ .  $\square$

Checking whether a graph  $G$  has a proper  $k$ -coloring is a  $(V, \mathcal{F}, k)$ -partition problem where  $\mathcal{F} = \{\mathcal{F}_1, \dots, \mathcal{F}_k\}$ , for all  $i = 1, \dots, k$ ,  $\mathcal{F}_i = IS(G)$ , the set of all independent sets of  $G$ . Note that Theorem 4 already implies a  $O^*(2^n)$  time and space algorithm for the Chromatic Number problem. An algorithm with the same time and space bounds was given recently by [10].

For  $k$ -coloring with preferences the families  $\mathcal{F}_i$  are defined slightly differently. Namely,  $\mathcal{F}_i$  consists of all independent sets  $I$  of  $G$  that can be assigned the same color without violating any list constraints and preferred color constraints:

- (list constraints):  $\forall v \in I, i \in L_v$ ;
- (preference constraints): for all  $1 \leq i \leq k$  such that  $v_i \in I$ ,  $N_G[u_i] \cap I \neq \emptyset$ .

Using this interpretation of the coloring with preferences problem, we give an algorithm solving the Chromatic Number problem on graphs of bounded average degree in time  $O^*((2 - \varepsilon)^n)$  and exponential space.

**Theorem 5.** *There is an algorithm checking whether for a given graph  $G$  of average degree  $d = O(1)$  there exists a proper  $k$ -coloring in time  $O^*((2 - \varepsilon(d))^n)$  and exponential space.*

*Proof.* First consider the case  $k \geq 2d$ . Note that  $|V_{\geq k}| \leq \frac{nd}{k} \leq n/2$ . Then one can find a proper  $k$ -coloring of the graph  $G[V_{\geq k}]$  in time  $O^*(2^{n/2})$ . Such a coloring can be easily extended to the whole graph (since there always exists a vacant color for a vertex of degree at most  $k - 1$ ). Thus, in the following we assume that  $k < 2d = O(1)$ .

Let  $\nu \geq 1, \mu < 1, a \geq 0, 0 < c < 1$  be constants to be defined later and let  $A, Y \subseteq V$  be as provided by Lemma 10. For  $Y$  we try all possible colorings in time  $k^{|Y|}$ . A fixed coloring of  $Y$  produces a list coloring problem for  $G \setminus Y$ . Let  $L$  be one of the most frequent color lists of vertices from  $A$ . Let  $C = \{v \in A : L_v = L\}$ . Since there are at most  $2^k$  different lists,  $|C| \geq |A|/2^k$ . If  $|C|$  is not divisible by 2 remove an arbitrary vertex from  $C$  and let  $C = \{u_1, v_1, \dots, u_t, v_t\}$ . Let now  $P = \{(u_1, v_1), \dots, (u_t, v_t)\}$  be a set of preferences.

It is straightforward to find a  $(p, q)$ -system of families with infants for the resulting partition problem. Let the number of families  $p$  be equal to  $t$ . The infant  $r_i$  of the  $i$ -th family is  $v_i$  and its family  $R_i$  is  $N[u_i] \cup \{v_i\}$ . This is a system of families with infants by definition (by choosing a small enough constant  $c$  one can guarantee that  $pq \leq n$ ).

To estimate the running time, we first recall that  $p = \frac{|C|}{2} = \frac{|A|}{2^{k+1}}$  and  $q \leq 2d + 1$ . Corollary 3 implies that the running time of the resulting algorithm is at most

$$k^{|Y|} \cdot \left( 2^{|V \setminus Y|} \cdot \left( \frac{2^q - 1}{2^q} \right)^p \right) \leq 2^n \cdot \left( k^{|Y|} \cdot \left( \frac{2^q - 1}{2^q} \right)^p \right).$$

We now choose the constants  $\nu, \mu, a$  so that (6) implies that the expression in parentheses is at most  $2^{-\beta n}$  for a constant  $\beta > 0$ . Let  $a = 0$ ,  $\nu = k$ , and

$$\mu = \left( \frac{2^q - 1}{2^q} \right)^{\frac{1}{2^{k+1}}}$$

(recall that  $k < 2d = O(1)$ ). It is easy to see that (6) then implies that the total running time is  $2^{(1-\beta)n}$  for a constant  $\beta > 0$ .  $\square$

## 6 The traveling salesman problem

**Theorem 6.** *The traveling salesman problem on graphs of average degree  $d = O(1)$  with integer weights from  $[M]$  can be solved in time  $O^*(M \cdot (2 - \varepsilon(d))^n)$  and polynomial space  $\text{poly}(n, \log M)$ .*

*Proof.* We construct a system of families with infants for the polynomial  $P$  given by (1). Let  $\mu < 1, 0 < c < 1$  be constants to be defined later. Let  $A, Y \subseteq V$  be provided by Lemma 10. Consider an optimal weight Hamiltonian cycle  $C$  in the graph. Let  $Y' \subseteq V$  be the successors of the vertices from  $Y$  in the cycle  $C$ . We guess the set  $A \cap Y'$  ( $O^*\left(\binom{|A|}{|Y|}\right)$  choices) and let  $A' = A \setminus Y'$ . Note that  $|A'| \geq |A|/2$  (since  $|A| \geq 2|Y|$ ) and for each vertex  $v \in A'$  its predecessor  $u$  in the cycle  $C$  belongs to  $V \setminus Y$ .

Let  $A' = \{1, \dots, p\}$ . Then for all  $i = 1, \dots, p$ ,  $R_i = N_{G \setminus Y}[i]$  and  $r_i = i$ . Clearly,  $|R_i| \leq q = 2d + 1$ . By choosing  $c < \frac{1}{2d+1}$  we can guarantee that  $pq \leq n$ . All  $R_i$ 's are disjoint since  $A'$  is an independent set in  $(G \setminus Y)^2$ . Finally, if the set  $Y'$  is guessed correctly (i.e.,  $Y'$  are indeed successors of  $Y$  in the optimal cycle  $C$ ) then  $((R_1, r_1), \dots, (R_p, r_p))$  is a  $(p, q)$ -system of families with infants for  $\mathcal{F}_P$ . Indeed, if  $r_i \in F$  for some closed walk  $F \in \mathcal{F}_P$  then two  $r_i$ 's neighbors in  $C$  must lie in  $V \setminus Y$ , i.e., in  $R_i$ .

By Theorem 4 the total running time does not exceed

$$\binom{|A|}{|Y|} \cdot 2^n \cdot \left( \frac{2^q - 1}{2^q} \right)^p \cdot M.$$

Recall that  $p = |A'| \geq |A|/2$ . Choose  $\mu = \left( \frac{2^{2d+1}-1}{2^{2d+1}} \right)^{1/2}$ . Then (6) implies that  $\binom{|A|}{|Y|} \cdot \left( \frac{2^q-1}{2^q} \right)^p < 2^{-\beta n}$  for a constant  $\beta > 0$ . Thus the total running time is  $O^*(M \cdot (2 - \varepsilon)^n)$ .  $\square$

## 7 Counting perfect matchings

The algorithm for counting perfect matchings shares some common ideas with the algorithm for the traveling salesman problem presented in Theorem 6.

**Lemma 12.** *Let  $G$  be a graph of average degree  $d = O(1)$ . Then  $\bar{G}$  (the complement of  $G$ ) contains a matching consisting of  $\frac{n}{2} - O(1)$  edges.*

*Proof.* Clearly  $|V_{\geq \frac{n}{3}}| \leq \frac{nd}{n/3} = 3d$ . After removing all these vertices from  $G$  we get a graph with at least  $n - 3d$  vertices such that the degree of each vertex is at most  $\frac{n}{3}$ . This implies that the degree of each vertex in the complement of this graph is at least  $n - 3d - 1 - \frac{n}{3}$ . This is at least  $\frac{n}{2}$  for large enough  $n$ . By Dirac's theorem [14] this graph is Hamiltonian and hence contains a perfect matching.  $\square$

**Theorem 7.** *The number of perfect matchings in a graph  $G$  with  $2n$  vertices of average degree  $d = O(1)$  can be found in time  $O^*((2 - \varepsilon(d))^n)$  and polynomial space.*

*Proof.* Assume that the vertices of  $V = \{1, \dots, 2n\}$  are numbered in such a way that

$$(1, n+1), (2, n+2), \dots, (m, n+m) \notin E \quad (9)$$

where  $m = n - O(1)$ . Such a numbering exists due to Lemma 12 (and can be efficiently found since we can find a maximum matching in  $\bar{G}$  in polynomial time). Following [6] we reduce the problem of counting perfect matchings to a problem of counting cycle covers of a special type. Construct an auxiliary multigraph  $G' = (V', E')$  where  $V' = \{1, \dots, n\}$  and each edge  $(i, j) \in E$  is transformed into an edge  $e = ((i \bmod n) + 1, (j \bmod n) + 1) \in E'$  with the label  $l(e) = \{i, j\}$ . In other words, we contract each pair of vertices  $(1, n+1), \dots, (n, 2n)$  and on each edge we keep a label showing where it originates from. Any two vertices in  $G'$  are joined by at most 4 edges. The average degree of  $G'$  is at most  $2d$ .

Recall that a cycle cover of a multigraph is a collection of cycles such that each vertex belongs to exactly one cycle. In other words, this is a subset of edges such that each vertex is adjacent to exactly two of these edges (and a self-loop is thought to be adjacent to its vertex twice).

An important property of the graph  $G'$  is the following: each perfect matching in  $G$  corresponds to a cycle cover  $C \subseteq E'$  in  $G'$  such that  $\cup_{e \in C} l(e) = V$  and vice versa. Indeed, each vertex  $i$  in  $G'$  is adjacent to exactly two edges. These two edges have different labels so they correspond to edges in the original graph  $G$  that match both  $i$  and  $i + n$ .

We have reduced the problem to counting cycle covers with disjoint labels in  $G'$  (the reduction is due to [6]). We further reduce the problem to counting cycle covers without self-loops. Note that by (9),  $G'$  has at most  $s = O(1)$  self-loops. For each such loop  $e = (i, i)$  we can consider two cases: to count the number of cycle covers with  $e$  we count the number of cycle covers in  $G'$  without the vertex  $i$ ; to count the number of cycle covers without  $e$  we can just remove the loop  $e$  from  $G'$  and count the number of cycle covers. This way, we reduce the problem to  $2^s = O(1)$  problems of counting cycle covers in a multigraph without self-loops.

We will count the number of cycle covers with exactly  $t$  cycles for each  $t = 1, \dots, n$  separately. For this, we define a polynomial  $P(x_1, \dots, x_n)$  containing a monomial for each closed walk with disjoint labels on adjacent edges (excluding the empty cycle) and compute the coefficient of the



monomial  $x_1 \dots x_n$  in  $P^t$ . This coefficient divided by  $t!$  is exactly the number of cycle covers with  $t$  cycles.

To evaluate  $P$  efficiently we use dynamic programming again. Let  $Q_k(x_1, \dots, x_n)$  be equal to the sum of all monomials corresponding to all closed walks of length  $k$  with disjoint labels. Let also  $T_{u,v,l_0,l,k}(x_1, \dots, x_n)$  be equal to the sum of all monomials corresponding to all walks with disjoint labels of length  $k$  from  $u$  to  $v$  such that the label of the first edge is  $l_0$  and the label of the last edge is  $l$ . Note that we only need to ensure that the labels on adjacent edges are disjoint. Namely, one of the edges adjacent to a vertex  $i$  in  $G'$  must contain  $i$  in its label while the other one must contain  $i + n$ . Then

$$T_{u,v,l_0,l_1,k}(x_1, \dots, x_n) = \sum_{l_2, e=(w,v) \in E' : l_1=l(e), l_1 \cap l_2 = \emptyset} T_{u,w,l_0,l_2,k-1}(x_1, \dots, x_n) \cdot x_u$$

$$Q_k(x_1, \dots, x_n) = \sum_{e=(u,v) \in E'} \sum_{l_1 : l_1 \cap l(e) = \emptyset} \sum_{l_2 : l_2 \cap l(e) = \emptyset} T_{v,l_1,u,l_2,k}(x_1, \dots, x_n)$$

The system of families with infants for  $\mathcal{F}_P$  is constructed in exactly the same way like in Theorem 6.  $\square$

## References

- [1] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in p. *Annals of mathematics*, pages 781–793, 2004.
- [2] Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2):308–340, 1991.
- [3] Eric Bax and Joel Franklin. A permanent algorithm with  $\exp[\Omega(n^{1/3}/2 \ln n)]$  expected speedup for 0-1 matrices. *Algorithmica*, 32(1):157–162, 2002.
- [4] Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *J. ACM*, 9:61–63, January 1962.
- [5] Andreas Björklund. Determinant sums for undirected hamiltonicity. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS’10, pages 173–182, 2010.
- [6] Andreas Björklund. Counting perfect matchings as fast as Ryser. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA’12, pages 914–921. SIAM, 2012.
- [7] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. The travelling salesman problem in bounded degree graphs. In *Automata, Languages and Programming*, volume 5125 of *LNCS*, pages 198–209. Springer Berlin / Heidelberg, 2008.
- [8] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Narrow sieves for parameterized paths and packings. *arXiv preprint arXiv:1007.1161*, 2010.
- [9] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Trimmed Moebius inversion and graphs of bounded degree. *Theory of Computing Systems*, 47(3):637–654, 2010.

- [10] Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set partitioning via inclusion-exclusion. *SIAM Journal on Computing*, 39(2):546–563, 2009.
- [11] Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Fast hamiltonicity checking via bases of perfect matchings. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 301–310, 2013.
- [12] Marek Cygan and Marcin Pilipczuk. Exact and approximate bandwidth. *Theoretical Computer Science*, 411(40–42):3701–3713, 2010.
- [13] Marek Cygan and Marcin Pilipczuk. Faster exponential-time algorithms in graphs of bounded average degree. In *Automata, Languages, and Programming*, volume 7965 of *Lecture Notes in Computer Science*, pages 364–375. Springer Berlin Heidelberg, 2013.
- [14] Gabriel A. Dirac. Some theorems on abstract graphs. *Proceedings of the London Mathematical Society*, 3(1):69–81, 1952.
- [15] Alexander Golovnev, Alexander S. Kulikov, and Ivan Mihajlin. Families with infants: a general approach to solve hard partition problems. *CoRR*, abs/1311.2456, 2013.
- [16] Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962.
- [17] Taisuke Izumi and Tadashi Wadayama. A new direction for counting perfect matchings. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 591–598. IEEE, 2012.
- [18] Ioannis Koutis and Ryan Williams. Limits and applications of group algebras for parameterized problems. In *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5–12, 2009, Proceedings, Part I*, volume 5555 of *LNCS*, pages 653–664. Springer Berlin / Heidelberg, 2009.
- [19] Leopold Kronecker. Grundzüge einer arithmetischen theorie der algebraischen grössen. *J. Reine Angew. Math.*, 92:1–122, 1882.
- [20] Daniel Lokshtanov and Jesper Nederlof. Saving space by algebraization. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 321–330. ACM, 2010.
- [21] Johan M. M. Rooij, Hans L. Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In Amos Fiat and Peter Sanders, editors, *Algorithms — ESA 2009*, volume 5757 of *Lecture Notes in Computer Science*, pages 566–577. Springer Berlin Heidelberg, 2009.
- [22] Herbert John Ryser. *Combinatorial mathematics*. Mathematical Association of America Washington, DC, 1963.
- [23] Rocco A. Servedio and Andrew Wan. Computing sparse permanents faster. *Information Processing Letters*, 96(3):89–92, 2005.
- [24] Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2 edition, 2009.